

Desktop to Cloud Migration of Scientific Computing Experiments

Satish Narayana Srirama¹, Pelle Jakovits¹ and
Vladislav Ivaništšev²

¹Mobile & Cloud Lab, Institute of Computer Science, University of
Tartu, J. Liivi 2, Tartu, Estonia

²Institute of Chemistry, University of Tartu, Ravila Str. 14A,
Tartu, Estonia

{srirama, jakovits, vladislav.ivanistsev}@ut.ee

November 26, 2015

Abstract

Scientific computing applications usually need huge amounts of computational power. The cloud provides interesting high-performance computing solutions, with its promise of virtually infinite resources on demand. However, migrating scientific computing problems to clouds and the recreation of software environment on the vendor-supplied OS and cloud instances is often a laborious task. It is also assumed that the scientist who is performing the experiments has significant knowledge of computer science, cloud computing and the migration procedure, which is often not true. Considering these obstacles, we have designed a tool suite that migrates the complete software environment directly to the cloud. The developed desktop-to-cloud-migration (D2CM) tool supports transformation and migration of virtual machine images, reusable deployment description and life-cycle management for applications to be hosted on Amazon Cloud or compatible infrastructure such as Eucalyptus. The paper also presents an electrochemical case study and computational experiments targeted at designing modern supercapacitors. These experiments have extensively used the tool in drawing domain specific results. Detailed analysis of the case showed that D2CM tool not only simplifies the migration procedure for the scientists, but also helps them in optimizing the calculations and compute clusters, by providing them a new dimension – cost-to-value of computational experiments.

1 Introduction

Scientific computing is a field of study that applies computer science to solve typical scientific problems. It is usually associated with large-scale computer modeling and simulation and often requires a large amount of computer resources. Cloud computing [44] suits well in solving these scientific computing

problems, with its promise of provisioning virtually infinite resources. Cloud computing is a style of computing in which, typically, resources scalable on demand are provided "as a service (aaS)" over the Internet to users who do not need to have knowledge of, expertise in, or control over the cloud infrastructure that supports them. The provisioning of the cloud services can be at the Infrastructural (IaaS), Platform (PaaS) or Software (SaaS) levels.

Cloud computing is intertwined with its three main features: elasticity, utility computing, and virtualization. A cloud platform dynamically provisions, configures, reconfigures, and deprovisions computing resources on demand and in real-time. The elasticity of a framework can be defined as its ability to adjust according to the varying loads of requests or requirements it has to support. Elasticity and the cloud's promise of provisioning virtually infinite resources make it ideal for resource-intensive scientific computing tasks. Utility computing model forwards the idea of computing as a utility, where consumers pay based on how many resources they use. This allows the scientists to have no upfront commitment costs and allows to estimate the cost of experiments before actually proceeding with them. Last but not the least of the features is the virtualization [14] technique, which is used to separate the hardware and software and allows running several independent virtual servers on a single physical device.

Significant research [17, 58, 62] has been performed in migrating scientific computing applications to the cloud. A number of science clouds [30, 35, 56] have been established in recent years, in which scientists have run many scientific simulations and applications and measured their performance to evaluate the applicability of doing science on the cloud. However, the idea of running scientific computing applications on the cloud has not been well received by all quarters of the scientific computing community, as the performance of cloud still lags behind when compared to grids or clusters.

This is because communication and other types of latency added by the virtualization technology are major hindrances when executing scientific computing applications on the cloud [18, 58]. The cloud computing community is trying to address these issues, and several solutions have already been proposed [18, 42], in recent years. The support of high-performance computing (HPC) with Cluster Compute Instances by the public clouds like Amazon EC2 is also a positive step in this direction.

While the research of migrating scientific computing experiments to the cloud shows great potential, most often it is assumed that the users who port their applications to the cloud have significant knowledge of computer science, cloud computing, and the migration procedures. However, from our observations this is not the case with non-computer scientists. For example, scientists who are running their computational modeling and simulation tasks on grids and clusters are unaware how the environment is configured, how the queues work on clusters and how their jobs get executed. All they are interested in is that they submit a job to some queue and after some time they can collect the results. Having this assumption in mind, we have designed Desktop-to-Cloud-Migration (D2CM) [55, 57] tool suite that helps scientists to migrate their experiments to the cloud. The idea is to migrate the complete software environment, in which the scientists have set up their experiments, directly to the cloud.

D2CM tool integrates several software libraries to facilitate VM image transformation, migration, and management of computing experiments running in the

cloud. To be able to leverage the cloud resources, we specifically concentrate on migrating distributed applications, such as applications which use the Message Passing Interface (MPI) to parallelize the execution of the computations, but it can be used for any distributed solution or standalone application. The tool also supports monitoring the health of the cluster while the application is running in the cloud.

Once the prototype of the D2CM tool was ready, we used it to migrate several of our scientific computing experiments to the cloud. This paper addresses our experience with migrating an electrochemistry case study to the cloud. Grid-base Projector-Augmented Wave Method (GPAW) [26] electronic structure calculator was used for the computing experiments. GPAW is a popular distributed program suit which in conjunction with the Atomic Simulation Environment (ASE) [13] provides optimized tool-set for electronic structure calculations. The GPAW program was used to study the capacitance of an interface between a metal electrode and a room-temperature ionic liquid. The computing experiments we performed with the aim to improve supercapacitors' and batteries' performance [38] are explained briefly along with their field-specific necessity. In the process, we also discuss the experiments performed by the electrochemists to show how they could scale their trials by successfully exploiting the cloud resources and characteristics, using the D2CM tool.

The rest of the paper is organized as follows. Section 2 describes the cloud migration procedure in detail along with the developed tool. Section 3 briefly explains the considered electrochemical case study while the conducted computing experiments and some of the observed results are summarized in section 4. Section 5 provides an overview of the related work. Section 6 concludes the paper with future research directions.

2 Desktop to Cloud Migration

A major hurdle in migrating applications to an IaaS cloud is the re-creation of a software environment on the vendor-supported OS and cloud instances. The D2CM tool seeks to mitigate potential problems of this often laborious task by taking the entire environment as-is, provided it is within a virtual environment. The D2CM tool supports non-interactive applications which operate on a finite input data set and it provides three main facilities:

- Programmatical transformation of user-supplied virtual machine images to the target cloud image format. Currently, it has support for migrating VirtualBox disk images (VDI) to Amazon Machine Image (AMI) format and to Eucalyptus [27] compatible XEN [14] images.
- Life-cycle management of distributed applications, hosted on Amazon Elastic Compute Cloud (EC2) [8] platform or compatible infrastructure, such as Eucalyptus.
- Creating deployment templates to specify how exactly the application is to be deployed on the cloud.

2.1 Virtual image transformation

EC2's and Eucalyptus' underlying machine virtualization is XEN (Note: Eucalyptus also supports KVM), a widely used open-source technology which supports full and para-virtualization. While Amazon provides a range of already configured images, it also supports user-supplied images, provided they function with a Xen compatible kernel. The kernel must have pv-ops with XSAVE hypercall disabled [11]. An image may either be configured to boot with Amazon supplied kernels, known as AKIs, or with a kernel packaged within the image itself. The later option is used by D2CM, which uses Xen's PVGRUB boot process. In the kernel update section of the transformation process, the D2CM tool installs a new kernel, either 32 or 64 bit depending on the detected image type, and modifies GRUB configuration to boot with the new kernel by default. In the current implementation, the tool only supports the transformation of a single disk partition. If an image has more than one partition, only the primary one can be migrated to EC2.

D2CM uses the Libguestfs library [2] to inspect and manipulate virtual machine images. Using libguestfs is highly preferred to mounting the images within the user's filesystem as directly mounting the image requires the host OS to support the image's filesystem(s) and requires root privileges. Libguestfs' API includes functions to extract partitions from images and to insert SSH key files, which are required during the image migration process. Libvirt [3] is an abstraction library for interacting with desktop hypervisors. The D2CM tool uses libvirt to access the currently supported hypervisor, VirtualBox, which is an open source hypervisor that runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems. In the future, adding support for other popular hypervisors such as KVM, Xen, and VMWare should be greatly eased by the use of libvirt. The currently supported IaaS platforms, EC2 and Eucalyptus, provide the same Amazon Web Services (AWS) API for managing machine instances. The web service is wrapped by several libraries which provide programming language specific bindings. As the D2CM tool is developed using Python, we used the Boto library which is the Python binding for AWS.

In addition to working with a compliant kernel, an instance created from this image must also be able to mount devices in the manner supplied by the EC2 virtual machine. EC2 instances have a default device mapping that usually does not match the fstab entries within the user-supplied image. EC2 instances come in a variety of types. Some instances do not provide swap devices, while, most instances provide additional storage devices that the user may want to utilize. To conform to this new virtual hardware environment, D2CM modifies the fstab appropriately.

2.2 Virtual image migration

Before the transformation process, D2CM tool goes through the following steps to actually transform and migrate a user-supplied image.

- **Specifying cloud credentials:** The first step to application migration requires the user to supply the tool with valid AWS credentials so that the tool can access cloud services. For Amazon EC2 and Eucalyptus,

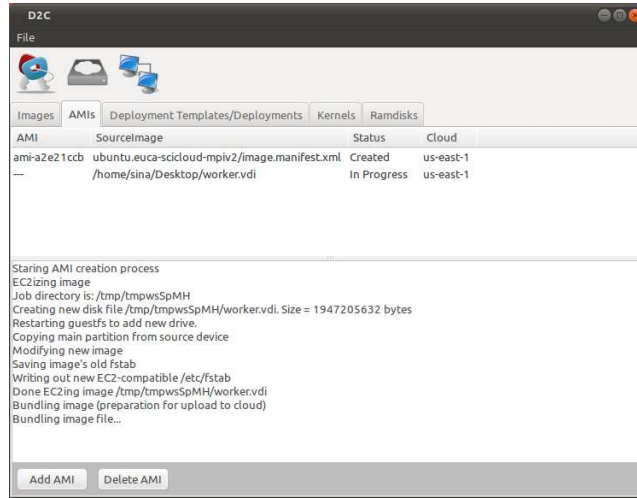


Figure 1: Image transformation view in the D2CM tool

these credentials include user id, access key, secret key, user certificate file location and private key file location.

- **Image registration and migration:** The user then registers a desktop image with the tool by choosing a name, an image file located in the local machine and the default cloud for this image. The user can then select to initiate the migration process, by choosing the target cloud, its availability region, and bucket name in the cloud file system, where the virtual machine image will be stored.
- **Image transformation:** The tool proceeds with the image transformation steps, making any changes to a separate image file, to preserve the integrity of the source image. (See figure 1)
- **Image upload and registration:** The tool uploads the transformed image to Amazon's Simple Storage Service (S3) [9] or to Walrus in the case of Eucalyptus. The tool then registers the image with the EC2/Eucalyptus, declaring default settings such as suitable system architecture, default kernel, and ramdisk, default instance type, etc. All images are registered by default as private, only authorizing read and boot access to the account which was used to upload and register the image.

The duration of the entire transformation and upload procedure can vary greatly depending on the size of the image and internet connection speed. At this point, the user's image is ready to run on the cloud. This process must be repeated for each unique desktop image required for the distributed application.

2.3 Defining deployment template

Next step after migrating the virtual machines is defining a deployment template that describes how the virtual machines are divided into roles and how these

The screenshot shows a web-based form for defining a role in the D2CM tool. The form is organized into several sections:

- Uploads:** Contains two rows. The first row has 'Source' set to `/home/pelle/experiments/ec2` and 'Destination' set to `/tmp/ec2.sh`, with a 'Remove' button. The second row has empty fields and an 'Add' button.
- Start Scripts:** Contains one row with the command `ssh-keyscan -f /tmp/d2c.context -t rsa >> ~/.ssh/known_hosts` and a 'Remove' button, followed by an empty row with an 'Add' button.
- Async Start Scripts:** Contains one row with the command `bash /tmp/ec2.sh` and a 'Remove' button, followed by an empty row with an 'Add' button.
- File Done Check:** Contains one row with the path `/tmp/done.txt` and a 'Remove' button, followed by an empty row with an 'Add' button.
- Data to Collect:** Contains one row with the path `/tmp/output.txt` and a 'Remove' button, followed by an empty row with an 'Add' button.
- Bottom Section:** Features three dropdown menus: the first is set to 'c1.medium', the second to '3', and the third has a '+' and '-' button. To the right of these are 'Save' and 'Cancel' buttons.

Figure 2: Defining a role in the D2CM tool

roles are executed on the cloud. A deployment consists of one or more roles and each role is defined by a name, association to a previously migrated cloud image, instance count, instance type and life-cycle scripting. The number of roles is not limited, and there may be a many-to-one relationship between roles and a unique cloud image. The instance count defines the number of virtual machines that will be allocated upon deployment start-up for a role. Roles include the following optional life-cycle commands that the deployment manager invokes at the appropriate stage:

- Input files that should be uploaded from the local machine to the virtual machines.
- Start-up command(s) that are executed when the virtual machine boots.
- Initiation commands that specify which command starts the asynchronous execution of the applications (in our case, computing experiments) inside the virtual machines.
- Ending condition for checking when the computing experiment has been completed.
- Download actions that specify which files are downloaded once the computing experiment is done.

Figure 3 illustrates a deployment template model that we used for executing scientific computing experiments in Amazon EC2. Additionally, figure 2 shows the actual user interface for defining the deployment roles in the D2CM tool.

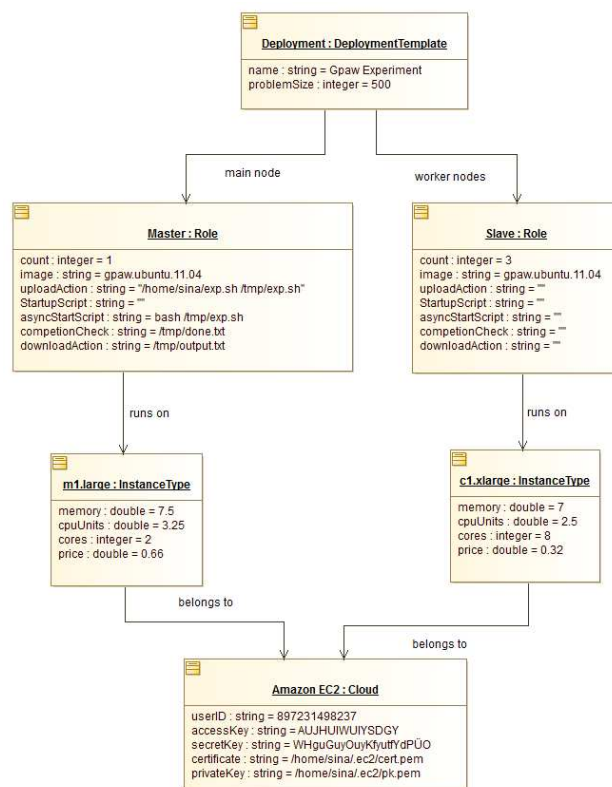


Figure 3: An example deployment template model

2.4 Executing a deployment on the cloud

D2CM supports non-interactive applications which operate on a finite input data set. The D2CM tool does not directly support interactive applications which may prompt for some kind of instructions or responses from the user. For such applications, the user can directly log into the cloud instances from their local machine via ssh.

The defined deployment template describes a specific type of experiments and to execute a single experiment on the cloud the user has to create a deployment instance from the deployments and initiate its execution. The started deployment and its life-cycle proceed through the following phases in the D2CM tool.

2.4.1 Reservation

Upon initiation, the tool iterates through deployment roles and reserves instances matching user specifications. If the request cannot be completely fulfilled by the cloud provider, any successful allocation is released and the process ends. AWS begins charging the user's account, the moment instances are reserved. Prior to the actual reservation, the tool prompts the user for confirmation, displaying the per hour rate in USD as determined by the deployment's instance types and count, to confirm that the user is prepared to pay this price.

2.4.2 Contextualization

Processes that operate cooperatively require some discovery mechanism and the tool facilitates this through contextualization step. When all instances within a deployment are assigned IP addresses, the tool makes sure every instance has access to all the IPs in the deployment, through configuration files.

2.4.3 Start-up

Once the instances have been acquired, the tool executes any user-defined start actions. If a role contains multiple instances, the start actions will be run on each instance. Currently, the tool supplies three classes of start-up actions: synchronous, asynchronous, and upload.

The upload action allows the user to install files on the instances which were not bundled with the image. These files may either correct minor image defects or supply computing experiment specific input. The synchronous action executes a specified BASH script. The tool serially executes all scripts for a given role and its instances. This functionality allows users to make modifications to deployment hosts that are guaranteed to finish before starting any main scripts, which are run using the asynchronous actions. The asynchronous action executes a script remotely on an instance, returning execution control immediately back to the tool. In contrast to the synchronous action, the user should assume that all asynchronous actions run in parallel. The D2CM tool logs on to the remote instance, executes these commands in a daemonizing (with *nohup*) wrapper and immediately ends the remote connection. This is the mechanism by which a user should initiate the main application process. An example asynchronous action is:

```
mpirun -np 16 python matrix_calc.py
```


Once all asynchronous actions have been initiated, the tool changes the deployment state to completion monitoring.

2.4.4 Completion monitoring

The user must configure the D2CM tool to test for application-specific completion. The only option currently provided is the ability to monitor for the presence of a user specified file on an instance. The tool continually polls the instances of roles with the specified check. If no check is provided for a given role, that role is automatically assumed to be in the finished state. The program proceeds to finalization only when all the roles are in the finished state.

2.4.5 Finalization

The result of supported applications will be a collection of output files. These must be migrated to persistent storage as the cloud instances only provide ephemeral storage, i.e. storage that exists for a limited life-span of the instance. D2CM supports configurable downloading of the user specified files from the instances to the local storage.

2.4.6 Termination

Upon retrieval of the required files, the deployment is shut down. D2CM terminates all instances associated with the deployment, ending any of the runtime charges from the cloud provider.

2.4.7 Recovery from failures

D2CM stores the running state of experiments in a local database and in a case of network or machine failures, it can reconnect to the cloud infrastructure and restore the state of the experiment. The user can shut down the tool while an experiment is running and continue to monitor it later, which is necessary for long-running experiments. This feature is critical especially as the experiments are performed from the desktop.

2.5 Results and re-execution of finished experiments

Once the experiment is completed, the tool provides a link to the results as shown on the figure 4. It points to a "work directory" on the local machine, where all the results (download actions specified in the deployment template) were downloaded, which can be accessed by the user to verify the results of the experiment.

D2CM tool keeps track of all the previously run experiments and the user can re-execute any of them by simply cloning the deployment instance, changing some parameters and deploying them. The user can also run multiple experiments simultaneously and the tool can keep track of all of them. This provides convenient means to execute a chain of experiments which only differ by some parameter values. It also helps to scale up the deployment infrastructure between experiments without having to create new deployment descriptions from scratch.

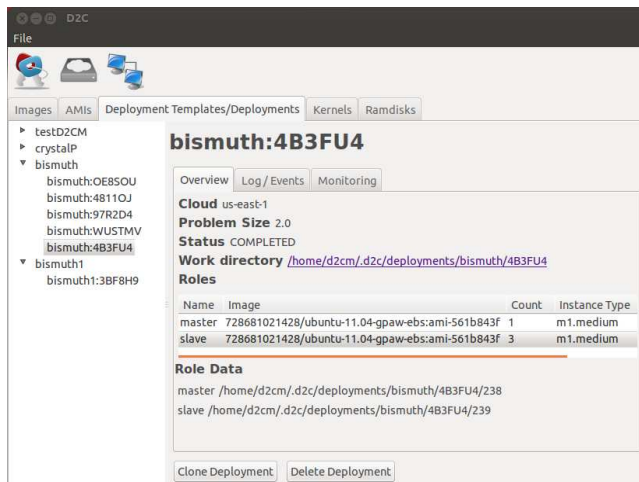


Figure 4: A completed deployment shown in the D2CM tool

3 Migration Case Study

The goal of this research is to provide effective tools for conducting computational experiments on a cloud. To validate the approach, we ported a computational electrochemistry case study to the cloud.

Electrochemistry is an actively studied branch of chemistry, and some of its most well known applied solutions are batteries, fuel cells, supercapacitors and solar panels. Several Noble prizes were awarded during the past decade for discoveries concerning electrochemistry. One of them was shared between Walter Kohn for his development of the “density-functional theory (DFT)” and John A. Pople for his development of “computational methods in quantum chemistry” [4]. Another was shared between Martin Karplus, Michael Levitt, and Arieh Warshel “for the development of multiscale models for complex chemical systems” [5]. The use of DFT-based computational methods to understand electrochemical processes by examining molecular scale models is continuously gaining popularity within the electrochemistry community [12, 19].

In the case study, we investigated room temperature ionic liquids [28] by the means of computational electrochemistry methods to find a better electrolyte for supercapacitors [22]. Ionic liquids are a novel type of electrolytes with a unique combination of properties that can be used to improve the performance of batteries, fuel cells, supercapacitors and solar panels [28]. Supercapacitors have several advantages over other energy storage and transformation devices: they have impressive charge–discharge time (less than 10 seconds); virtually unlimited number of life-cycles (over one million); high specific power (10 kW/kg); and excellent operating temperature range (−20 to 65°C) [41]. However, when compared to common lithium-ion batteries, their cost per $W \times h$ is ten times larger and the amount of energy they can store is ten times lower. The performance of capacitors and batteries is compared in figure 5, representing energy density vs. power density chart for various energy-storage devices. “Right and higher” is better in the chart; data for lithium-ion batteries are in the upper left corner, and data for supercapacitors are on the right.

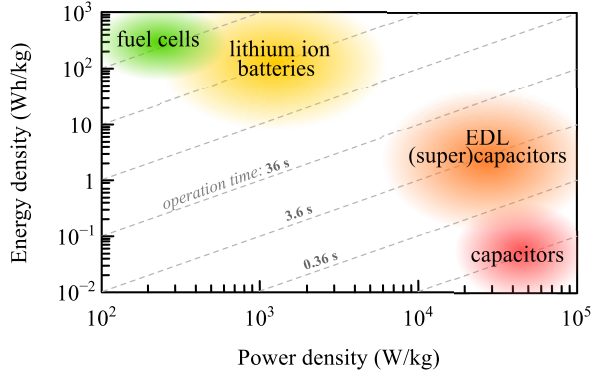


Figure 5: Comparison of lithium battery and supercapacitor

The energy density (E) of supercapacitors can be improved in accordance with the following equation:

$$E = \frac{1}{2} C \Delta U^2, \quad (1)$$

where C is capacitance and ΔU is voltage.

Currently, the voltage is almost the same for both lithium battery and supercapacitor, but with ionic liquids it can be increased for the later by two times from 3 Volts to almost 6 Volts [41]. Moreover, it is also possible to increase the capacitance. Herewith, the scientific challenge is to reach the best possible characteristics at the laboratory conditions. In this process, our computer modeling may hint at the direction for the experimental research. The engineering challenge is to improve characteristics of modern supercapacitors and simultaneously reduce their price by designing them in such a way, that they would be commercially competitive with modern batteries and thus would become available at the market. A successful research and development of novel prototypes have been conducted at the University of Tartu [40, 43], laying the ground for the world's leading high-performance supercapacitor producer Skeleton Technologies [54]. We contribute to this success by performing computational modeling of ionic liquids at electrode surface as described below.

3.1 Computational modeling

We started our computational study with a fairly simple model of Au(111) | EMImBF₄ interface by representing the interface as a *gold slab* (electrode surface) and a *single adlayer of ions* (figure 6) [38]. Figure 7 shows a model with a thick (2 nm) overlayer of an ionic liquid. This model is consistent with experimental results, which were obtained with the help of scanning tunneling microscope, electrochemical impedance spectroscopy and cyclic voltamperometry [34, 47]. These results suggest that at positive electrode potentials a structured layer of anions is formed; at negative potentials cations form an ordered layer structure, and in between, both anions and cations are present at the surface. The latter case requires a complex model, shown in figure 7, and was omitted in the presented study.

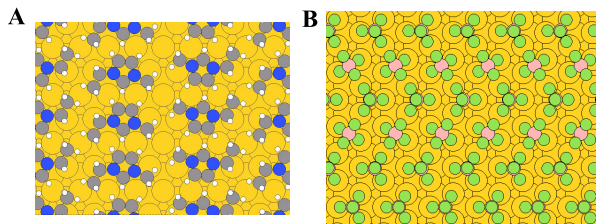


Figure 6: A. Molecular model for a cationic (1-ethyl-3-methylimidazolium) layer at negative Au(111) surface; B. Molecular model for an anionic (tetrafluoroborate) layer at positive Au(111) surface

We investigated the two preceding cases separately by performing DFT calculations with the help of GPAW tool [20,26], which is described in the following subsection in more detail. Starting from a given geometry, the GPAW calculator finds optimal geometry with the lowest energy and evaluates the electron density, which is then used to evaluate absolute electrode potential.

The model used (figure 6) assumes domination of only one type of ions (characterized by coverage – a variable number of ions per surface area unit) which completely compensate the charge at the electrode surface. According to the DFT formalism, the charge density is redistributed during the calculation procedure so that the electric field sets up between the surface and ions. In this way we get a simple molecular-scale capacitor, which to a certain degree mimics the real interface between an electrode and an ionic liquid. According to the Molecular Dynamics simulations results, at high positive and negative electrode potentials a dense structured monolayer of ions at electrode surface coexists with apparently non-structured ionic liquid further from the electrode [37,61]. This observation hints to a possible reference point between the models used in Molecular Dynamics simulations as well as in our DFT calculations and the real interface. We reasonably assume that at other potentials our models provide a good guess of interfacial characteristics. For qualitative comparison, in section 3.3, we focus on the interfacial free energy and Bader-type charges of ions.

According to the formula 1, a higher energy density can be achieved by increasing the capacitance and voltage. To do so, it is required to understand how these two parameters depend on structure and composition of the interface between ionic liquid and electrode. This goal can be achieved either experimentally or theoretically through computational modeling. Within the experimental approach, very complex systems are studied predominantly at macroscopic level, while the computational modeling operates with very simple systems represented by molecular-scale models. Therefore, the challenge of obtaining reliable results from computational experiments is correlated with balancing between the model complexity and the amount of available resources. A sophisticated model obviously accounts for more effects and thus is more realistic, but is also much more resource demanding.

To obtain most realistic results, which would be comparable to the experimental electrochemical results, one needs high-CPU resources, a migration tool, quantum chemical calculator and a reliable model. In our study, these are the cloud resources, D2CM, GPAW program suite, and the model justified above, respectively.

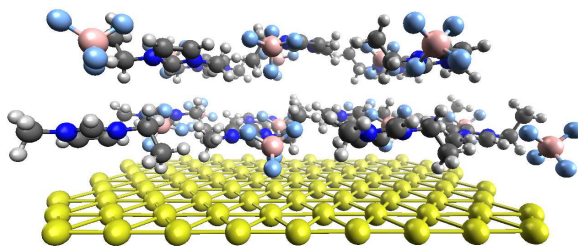


Figure 7: Molecular model of an interface between neutral Au(111) surface and 1-ethyl-3-methylimidazolium tetrafluoroborate

3.2 GPAW program

Computational routines within GPAW program are based on rather novel Grid-based Projector-Augmented Wave method [20,26]. The code is released under the GNU Public License (GPL) and has several distinguished features, which are necessary for electronic structure calculations of large electrochemical systems via parallel computing. Firstly, there are at least two ways of representing the wave functions: 1) using lcao-mode, which gives less accurate results, but computationally is more efficient and provides a good guess for optimized geometry; 2) using “finite difference” or plane-wave mode, which gives more accurate results, but is highly demanding on computational resources. Secondly, the parallelization with MPI can be done over the k -points, bands, spin in spin-polarized calculations, and using real-space domain decomposition. Finally, the time-dependent DFT implemented in GPAW program enables calculation of dielectric function, optical and X-ray adsorption spectra. In addition to vibrational spectrum, these in principle allow a direct comparison to experimentally measured data. With the main focus on the D2CM migration process, in this study we proceed with optimization of several systems, without performing any sophisticated time-dependent DFT calculations.

Setup for a single computational experiment is defined via the atomic simulation environment (ASE) python syntax [13], which is used to define atomic positions, calculator routines used, optimization procedures, input-output data etc. The following script is an example of one of our computational experiments:

```

1 #Define geometry of metal slab
2 slab = fcc111('Au', size=(2,2,3),
3             orthogonal=True)
4 slab.center(axis=2, vacuum=12)
5 d=0.8
6 a=asin(1/sqrt(3))+pi
7 #Define geometry of BF4 anion
8 BF4 = Atoms([Atom('B', ( 0, 0, 0)),
9             Atom('F', ( d, d, d)),
10            Atom('F', (-d,-d, d)),
11            Atom('F', (-d, d,-d)),
12            Atom('F', ( d,-d,-d))])
13 #Change orientation of BF4
14 BF4.rotate('y',pi/4,center=(0,0,0))
15 BF4.rotate('x',a,center=(0,0,0))
16 BF4.rotate('z',pi/3,center=(0,0,0))
17 BF4.translate(slab.positions[5]+(0,0,5))

```

```

18 #Put BF4 above the metal slab
19 slab += BF4
20 #Fix second and third layers in the slab
21 mask = [atom.tag > 1 for atom in slab]
22 slab.set_constraint(FixAtoms(mask=mask))
23 #Setup calculator
24 calc = GPAW(
25     h=0.16,          #grid spacing
26     kpts=(4,4,1),    #number of k-points
27     txt='22.txt',     #output file
28     parallel={'domain':4}, #parallelization
29     xc='RPBE')         #functional
30 slab.set_calculator(calc)
31 qn=optimize.QuasiNewton(slab,
32     trajectory='22.traj')
33 qn.run(fmax=0.05)

```

The script specifies a system, a calculator (GPAW) and an algorithm for geometry optimization (QuasiNewton). The model consists of a molecular fragment ($\text{BF}_4 = \text{Atoms}([\text{Atom}(\text{'B'}, (0, 0, 0)), \dots])$) and an infinite gold slab ($\text{slab} = \text{fcc111}(\text{'Au'}, \text{size}=(2,2,3), \dots)$). Two layers in the gold slab are kept fixed in their bulk positions ($\text{slab.set_constraint}()$). For the GPAW calculator we specify grid spacing ($h=0.16$) and number of k -points ($\text{kpts}=(4,4,1)$) at $4 \times 4 \times 1$ Monkhorst-Pack sampling grid used to perform Brillouin-zone integrations. In this script we use revised Perdew–Burke–Ernzerhof (RPBE) exchange–correlation functional, but we also tested a superior vdW-DF functional to take into account the dispersive (van der Waals) interactions [38]. A convergence criteria for the structural optimization according to QuasiNewton algorithm is set to 0.05 eV per Ångström for atomic forces ($\text{fmax}=0.05$).

3.3 Computational experiments performed and Results

In the conducted computational experiments, we used parameters similar to those shown in the script. Starting from a given geometry, the GPAW calculator was used to find an optimal geometry with the lowest energy. The obtained electron density was used to recalculate absolute electrode potential. The obtained energy values were used to estimate the interfacial free energy change via an approach introduced in Ref. [52]. Note, ΔG_{int} was calculated using the values consistent with the forces presented in the models and neglecting contribution of entropy terms.

The capacitance was determined from the interfacial free energy dependence on the electrode potential. Ideally, ΔG_{int} values should fit to a quadratic curve according to the equation 1: $\Delta G_{\text{int}} = \frac{1}{2} C \Delta U^2$, where ΔG_{int} is equal to the energy stored in an ideal capacitor which, in our case, is set up by ions and the counter charge on the gold surface. $\Delta U = (U - U_{\text{pzc}})$, where U is the electrode potential calculated from the work function ($U = W_e/e$) and U_{pzc} is the Potential of Zero Charge, taken to be equal to 5.3 V for our models [38].

Figure 8 illustrates the relation depicted in Equation 1 for the simplified interface models containing BF_4^- , F^- and I^- anions. The highest values for each of ions correspond to the formation of the ordered $\sqrt{3} \times \sqrt{3}$ adlayers (similar to the BF_4^- layer shown in Fig. 6). The obtained values for BF_4^- are in a reasonable agreement with the experimental values for high frequency capacitance: 5–10 $\mu\text{F}/\text{cm}^2$ [33, 34, 59] and for lower frequency capacitance at anodic peak 10–50

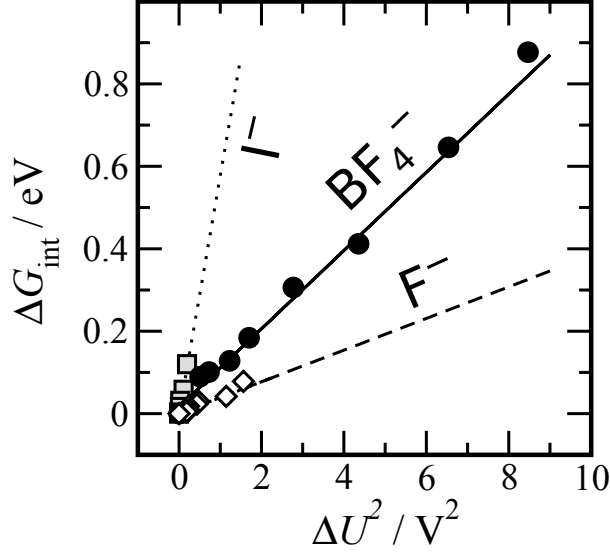


Figure 8: Dependence of interfacial free energy on electrode potential for molecular models of BF_4^- , F^- and I^- at Au(111) surface. The slope is proportional to the capacitance in case of an ideal parallel plate capacitor

$\mu\text{F}/\text{cm}^2$ [33, 34, 59] measured at gold surfaces.

The interfacial free energy is rising due to the repulsion between ions upon their accumulation near the surface. In the ΔG_{int} vs. U^2 representation the slope at the $\Delta G_{\text{int}}, \Delta U^2$ -dependence corresponds to the capacitance. For a hypothetical Au(111) | F^- interface, the evaluated capacitance value is lower when compared to the value for Au(111) | BF_4^- at the same electrode potential. That is a simple dependence on the composition of the ionic liquid. Herewith, among BF_4^- , F^- and I^- anion the iodide contributes to the highest capacitance at the interface.

Adlayers of iodide were observed in aqueous electrolytes and recently were found to form in ionic liquids [53]. As well as in the presented results for the simple model, at the laboratory conditions the formation of an adlayer is associated with the increase of capacitance. The difference between iodide and tetrafluoroborate ions is seen from Table 1 showing the partial ionic charge evaluated according to Bader charge analysis. The redistribution of electric charge densities are explicitly treated in DFT calculations, but until now were not accounted in modern theories and classical molecular dynamics simulations [23]. It is seen that iodide ions are involved in so-called partial charge transfer, when the absolute value partial ionic charge is reduced (in the case of iodide at low coverages to $-0.13 e$) due to the interaction with the surface. Oppositely, the charge on BF_4^- anion near the surface ($\approx -0.75 e$) remains almost as high as in neat EMIm BF_4 ionic liquid ($-0.9 e$). At very high potentials corresponding to high coverages iodide clearly transforms into a I^\cdot radical and BF_4^- dissociates according to the net reaction: $\text{BF}_4^- = \text{F}^- + \text{BF}_3$, where F^- has a lower partial charge and tend to react with the ionic liquid. The similar reaction mechanism was identified in recent X-ray photoelectron spectroscopy study [60].

To conclude, the conducted computational experiments shed light on the fol-

Coverage	0.05	0.125	0.333	0.375
$[\text{EMIm}]^{\delta+}$	0.79	–	–	–
$[\text{BF}_4]^{\delta-}$	0.85	0.75	0.69	0.61
$\text{F}^{\delta-}$	0.61	0.61	0.59	–
$\text{I}^{\delta-}$	0.13	0.13	0.13	0.03

Table 1: Partial ionic charge (δ / e) obtained via Bader-analysis of the all-electron density.

lowing aspects of charged interfaces in ionic liquids: 1) the capacitance values are determined by the strong repulsion between counter-ions; 2) high capacitance values are expected for ionic liquids containing iodide ions; 3) the stability of ionic liquids on the potential scale, at anode, is determined by electroreduction of anions. We note that the analysis of the results was conducted in close collaboration with the electrochemical group at the University of Tartu, in which the proposed aspects were verified in due course [53,60].

4 Evaluation of the migration process and the D2CM tool

Once the first version of D2CM tool was completed (and before the migration of the larger case study) we initiated a pilot project as a cooperation between computer scientists working with the D2CM tool and the scientists working in the electrochemistry field. Our primary goal was to validate the migration and deployment functionalities of the D2CM tool by verifying that the migrated systems are working as intended and that the results of the experiments are correct. We also aimed to make sure that the tool is fully usable by non-computer scientists.

We asked scientists working in the electrochemistry field to design a number of computational experiments that could be executed on the cloud using our tool. At the time of project initiation, corresponded to the state of the art in the field of electrochemistry. Table 2 summarizes a small subset of these experiments that have been conducted by us since 2010. Rows 1 to 4 describe the results of running a single experiment on varying cluster sizes and rows 5 to 8 describe randomly chosen computations in different scales.

It must be noted, that many of the current D2CM features were actually envisioned during this initial pilot project which involved intensive cooperation of the electrochemists and the D2CM developers. Once the pilot project was completed and the additional functionalities requested by the scientists were introduced, the consecutive experiments were performed by the electrochemists themselves without taking the help of computer scientists. While some of the results of these studies are already addressed in section 3, in the context of the case study, the more detailed results will be submitted to a domain specific journal [38].

One example of a D2CM feature that was introduced as result of cooperation with electrochemists is performance monitoring. Because electrochemical experiments (and especially the more complex ones) can run for extended peri-

Experiment ID	EC2 Instance type	Nodes (Cores)	EC2 Units	RAM (GB)	Time (Hrs)	Cost (USD)
dod1	m1.medium	1 (1)	2	3.75	3.10	0.48
dod2	m1.medium	2 (2)	4	7.5	1.60	0.48
dod4	m1.medium	4 (4)	8	15	0.87	0.48
dod8	m1.medium	8 (8)	16	30	0.42	0.96
22	m1.xlarge	1 (4)	8	15.0	4.3	3.80
22vdw	m1.xlarge	2 (8)	16	30.0	1.0	1.52
32	m1.xlarge	2 (8)	16	30.0	45.7	69.92
34	m1.xlarge	8 (32)	64	120.0	46.3	285.76

Table 2: A sample of GPAW experiments performed on EC2 using D2CM

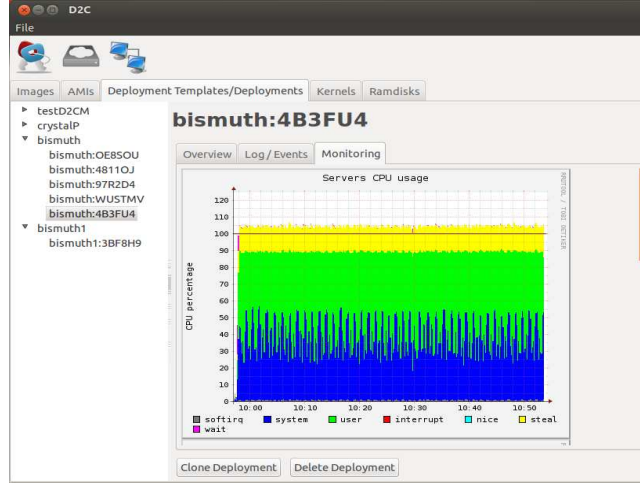


Figure 9: CPU profile of a computational experiment

ods of time the scientist needed a way to verify that the migrated experiment is working as intended and that the chosen deployment configuration was suitable. Thus we introduced real-time performance monitoring to the D2CM tool which allows the users to keep track of how well the computing resources are utilized.

Figures 9 and 10 show the CPU and memory profile of an computational experiment running on 4 m1.medium EC2 instances. D2CM uses `collectd` [46], an industry standard in distributed system monitoring, for extracting the performance parameters from the instances. The tool records CPU usage by type (user, system, idle and steal), memory usage, network I/O and system load for the whole deployed cluster. This feature can be turned off to save the system resources.

Figure 9 shows the average CPU usage of the hosts and the figure 10 depicts total memory usage across the hosts for one of the executed experiments. From these diagrams we could see that the CPU was utilized to the full capacity (with a significant amount of CPU steal) while memory was significantly underutilized. Thus it is clear that this experiment would significantly gain from a more capable CPU and that does not need as much memory. So the subsequent

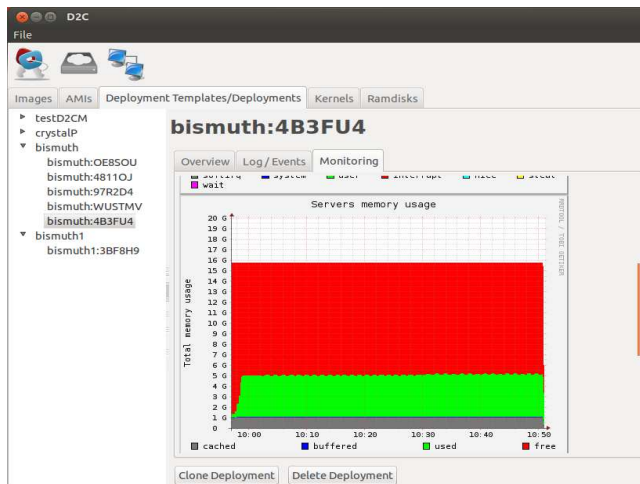


Figure 10: Memory profile of a computational experiment

computational experiments ran on High-CPU instance family of EC2, which have more powerful CPU and less memory than the standard instance family member m1.medium used in the previous computational experiment. Thus, the functionality that was initially introduced for keeping track of long running experiments also significantly helps in optimizing the deployed compute clusters, computational experiments’ set-up and the costs of the experiments.

In terms of costs of cloud computing resources, the usage of Amazon EC2 is still more justifiable for enterprise needs, while the usage of supercomputers or grids instead of clouds is still more suitable for academic users. For enterprise sector the cost of CPU-time at a supercomputer and EC2 is comparable. For instance, price per core hour at a regional supercomputer in UK is \$0.16 for a Dual Intel Xeon X5650 2.66 GHz CPU [36]. Price per core hour at EC2 is \$0.16 for Intel Xeon X5570 (4-core Nehalem architecture; Cluster Compute Quadruple Extra Large Instance) [7]. On the other hand, for typical scientific computing environments, there is no long queue to access shared infrastructure with D2CM. Thus, we may expect that in the near future, the D2CM tool will be used more often for academic purposes. Procuring and provisioning cost-effective computation cycles is possible via D2CM through optimizing the compute clusters; improving scalability by adjusting GPAW set-up; and utilizing Amazon EC2 Spot Instances. Overall, D2CM provides a good basis for effective high-performance calculations in electrochemistry as well as in the other scientific fields.

4.1 Scalability of computational experiments on cloud

To validate that the D2CM tool is able to utilize the distributed cloud resources in an efficient manner and to illustrate its usability for running distributed scientific experiment we also performed scalability tests. For measuring the scalability of the deployed system we once again used GPAW based experiments because GPAW supports a number of different means for defining the parallelism. It relies on the MPI for parallelization and can be rebuilt with opti-

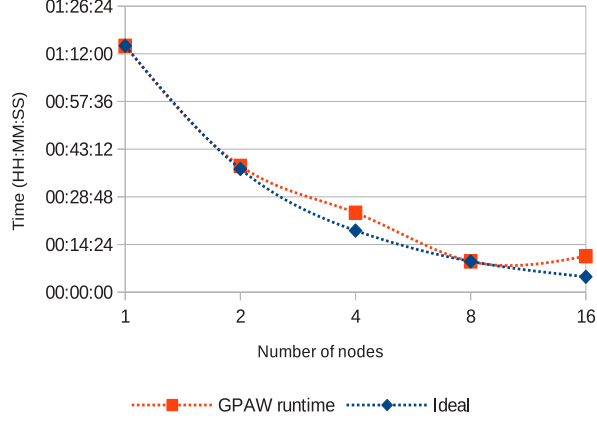


Figure 11: Runtime diagram of a GPAW scalability test

mized libraries for parallel computing such as Scalable Linear Algebra PACKage (ScaLAPACK) [15].

We chose a medium-size computational experiment and measured its runtime when deployed to EC2 cloud on clusters from 1 to 16 cloud m1.medium instances with “moderate” network performance (as assessed by AWS), 3.75 GB of memory and 2 EC2 Compute Units (1 virtual core). EC2 computing unit represents the capacity of allocated processing power and 1 EC2 unit is considered to be equivalent to an early-2006 1.7 GHz Xeon processor.

We measured the runtime of 20 optimization steps. The results are illustrated on figures 11 and 12. Figure 11 shows a runtime of a GPAW experiment when deployed on a varying cluster size from 1 to 16 nodes. The ideal curve is calculated assuming perfect scaling, which is possible with a high degree of software parallelization and excellent quality of the network. Figure 12 provides the parallel speedup, which shows how much gain do we get from introducing additional nodes. While this experiment does not gain any benefit from using 16 nodes over 8 nodes (showing the problem size might be too small to be deployed on 16 nodes), it does demonstrate that a virtual cloud cluster with moderate network performance provides good parallelization performance up to 8 nodes.

Some parallelization techniques are common among a number of program packages for performing electronic structure calculations, like GPAW, WIEN2k [16], Quantum Espresso [31], FEFF9 [50] as well as many others, for example, those sharing Libxc-library and listed in reference [45]. For this reason it might be expected that similar scalability, as seen in tests with GPAW, can be expected for the listed codes. Scalability tests made by Jorissen, Vila and Rehr for WIEN2k and FEFF9 codes complement our observation that a virtual cloud cluster provides parallelization performance as a physical cluster with very low latency and high bandwidth of the cluster interconnects [39]. Moreover, Jorissen et al. indicated a case when the network performance of regular EC2 instances becomes an issue, as in our test [39]. The problem arises due to the MPI intensive communication needed for large dense matrix diagonalizations employing the ScaLAPACK library [15]. On the cloud side it can be resolved by uti-

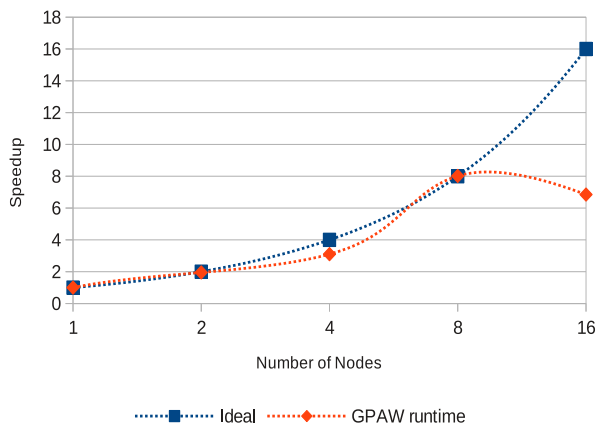


Figure 12: Speedup diagram of a GPAW scalability test

lization of HPC EC2 instances placed into a non-blocking 10 Gigabit ethernet network. In general the problem persists for systems with a large number of electrons. Therefore, we conclude that virtual cloud clusters provide a good speedup performance suitable for performing small to medium computational electrochemistry experiments.

4.2 Usability analysis of the D2CM tool and the cloud migration

Apart from the technical evaluation of the D2CM tool based migration procedure, we are also interested in its usability. In general, for research groups having a permanent heavy load of computational tasks, a private cluster is more justified than a cloud from the cost point of view. However, for small and medium research groups with lower intensity in computing and only basic expertise in HPC, it makes better sense to pay for resources and support on demand, from the electrochemists' perspective. D2CM tool provides an easy way for this cloud migration and it simplified running the actual computational experiments for the scientists, by allowing them to change the deployment configurations and computational experiment parameters on the fly. Feedback from the scientists who have been working with the tool show that they are quite satisfied with the tool and the migration procedure.

In addition to having a tool for cloud migration, to take full advantage of the cloud a scientist may also have to consider the following aspects. Firstly, the cloud solution can be used solely or to augment the available facility (cluster, GRID or HPC), e.g. when the facility is overloaded. Secondly, a large number of universities and companies have neither an in-house cluster nor the resources to build and maintain one. Access to the local GRID or HPC facilities might be limited due to a queuing system, specific restrictions, overload, etc. In such circumstances cloud provides access to a much larger number of CPUs that would usually be available to solve problems that are time-critical. Finally, one can use the cloud to run software that requires a different environment than

the one existing on the available facility. With the D2CM tool it is easy and quick to build the cloud environment and start using it, as if it was a desktop environment. Moreover, the tool also gives the extra flexibility at keeping track of the running costs and making them as low as possible.

The most obvious competitive edge of the cloud solution is its ability to explore specific research tasks in a timely, scalable manner and helps scientist to make better decisions quickly. Let us stress that a cloud can serve to augment existing clusters when the load on the cluster is too high. This way, the pay-per-use model can positively change the dynamics of computations in a research group. Even despite the higher price for the CPU time in common clouds, it can be very useful when time-to-result is a priority.

However, with the cloud migration, the type of computation and the software should be deliberately considered. On the same number of nodes there might be a large number of simultaneous small calculations or a single massively parallel calculation. From the practical point of view, the cloud computing is more efficient in the first case, but might struggle to fulfill a massively parallel job due to low interconnectivity between the nodes [58]. Next, precise estimation of time and the cost is only possible in some types of calculations, for instance classical molecular dynamics simulations [25]. It is a challenging, yet solvable task to estimate the costs for example for quantum chemical calculations, like in the presented computational electrochemical study.

Considering all these advantages and the successful migration of the electrochemistry case using the D2CM tool, currently discussions are underway to make the tool available to the GPAW community. The tool will soon be put on the GPAW site [20]. The tool is already released with a GPL License [55].

5 Related Work

D2CM tool provides three core features: migrating desktop virtual machines to IaaS platforms, specifying deployment configurations and life-cycle management of deployed system. While these features are present to some extent in other tools, D2CM is the only tool, as far as we know, that supports the whole process from migration to deployment and monitoring in one package and is specifically aimed for performing distributed scientific experiments in the cloud.

The main reason for developing the D2CM tool is that the migration of desktop images to the cloud has not been a widely supported feature by the cloud industry in the past. However, several tools have been recently published that do aim to provide this service. Amazon has developed VM Import [10] tool for migration of VMware ESX VMDK, Citrix Xen VHD and Microsoft Hyper-V VHD images to Amazon EC2 cloud. In addition to only supporting images from select proprietary systems, the VM Import tool currently supports only images running Microsoft Windows Server. Amazon has stated their commitment to add more supported platforms, but a concrete roadmap has not been published yet.

Racemi Cloud Path [49] is a solution that provides tools for migrating existing server environments, that run Red Hat Enterprise Linux or CentOS operating systems, to Amazon EC2, GoGrid, Rackspace, and Terremark public clouds. It automates the otherwise manual process of creating a virtual machine (VM) from an existing physical machine configuration and migrating VM

to the cloud. Racemi Cloud Path is not a free service and while it simplifies the migration process, it lacks the support for custom deployment configurations for the migrated virtual machines. Also, setting up already migrated environments has to be done manually, which can be a huge amount of work depending on the complexity of the deployment, especially when the number of different interacting roles and instances is high.

CloudSwitch [21] provides migration and deployment of virtual machines to the cloud and allows to integrate migrated servers with an existing system that runs locally or in other clouds. It is designed for setting up (or extending) longer running static server deployments. Defining and configuring different roles and instance configurations must be repeated each time a new deployment is executed. In comparison, once a deployment template (consisting of any number of roles, life-cycle actions, and instances) is defined in D2CM, the user can instantiate a new deployment, change its configuration parameters and then launch it, all in only a few seconds.

RightScale offers means of managing systems via deployment definitions, which make use of RightScale ServerTemplates [51]. ServerTemplate defines the machine image and scripts executed during the startup and shutdown phases of a machine. Amazon also provides EC2 system templating through the CloudFormation [6] tool. CloudFormation allows users to describe the images and machine types that comprise the logical roles of a system. While CloudFormation allows for reliable and somewhat flexible deployment of a system, any life-cycle management must be explicitly programmed into the VM images themselves. On Linux, this is supported by Amazon through the CloudInit software package. This conflicts with our approach in D2CM, where we seek to minimize the modification of the source images by the user, as our targeted users are pure scientists who are not proficient in managing server environments.

Similarly, other cloud frameworks that provide capabilities for automatic provisioning, deployment and monitoring of cloud systems are Cloudify [32], Puppet [48] and Chef [1]. However, while they are good for deploying more static systems on the cloud and might be suitable for setting up more permanent scientific clusters, they are not suitable for running scientific experiments with varied duration and they also do not provide means to define the life-cycle of such experiments for their automatic execution.

VIRTU framework [24] is a framework for configuration, deployment, and management of the life-cycle of distributed virtual environments. It provides users the ability to manage and configure application stacks on such environments by assembling building blocks consisting of operating system and application together with installation and configuration files. This framework is interesting for us because it deals with the deployment of distributed environments that large-scale scientific computing experiments most often require, with the main difference that it is not designed for Cloud, but rather for deploying virtual machines and applications on existing hardware.

Cloud Modelling Framework (CloudMF) [29] is a model-based framework being developed at SINTEF (<http://www.sintef.no>) for modeling the provisioning of cloud resources for complex software systems and their deployment and control in multi-cloud platforms. CloudMF provides a tool-supported domain-specific modeling language CloudML to model the provisioning and deployment of multi-cloud systems and a models@run-time environment for enacting the provisioning, deployment, and adaptation of these systems. The deployment of

the underlying modeling language CloudML has been partially supported by three European FP7 projects (REMICS, MODAClouds and PaaSage), where we were also involved, which all try to leverage its potential for simplifying the deployment and control of cloud applications.

While the modeling approach taken by CloudML can be very useful in the software engineering field and can really benefit the adoption of cloud platforms for more complex and especially legacy systems, the aim of the D2CM tool is to simplify the migration and deployment of scientific experiments for scientists who often have no prior knowledge of software engineering and modeling techniques. Still, we are investigating whether we can utilize CloudMF internally to streamline the deployment of scientific experiments in multi-cloud platforms while hiding all the technical details from the users. Our future research in this domain will try to address this issue.

6 Conclusions and future work

This paper discusses the direct migration of scientific computing applications to the cloud using the D2CM tool. The main goal is to migrate the complete software environment, in which the scientists have set up their computational experiments, directly to the cloud. The tool also facilitates and simplifies the execution of the experiments in the migrated system for non-computer scientists. The design and development of the D2CM tool is explained in detail. In the paper we also present an electrochemical case study targeted at designing modern supercapacitors. We discuss the role of D2CM tool in achieving domain specific results, the experiments performed, as well as the cloud scalability.

From the analysis of the case study, we observed that D2CM tool not only helps in the migration process but also simplifies deploying MPI applications in the cloud, helps to optimize the calculations, compute clusters and the costs of the experiments. Feedback from the users has shown that the D2CM tool also simplifies the actual experiment execution process by allowing the user to change the deployment configurations and experiment parameters on the fly. The tool and the simplified migration procedure can help the domain scientists avoid long queues at the grid resources and provide them a new dimension, cost-to-value of the computational experiments.

Our current work involves extracting scaling parameters for the calculations and using them to predict the cost of future experiments. The pricing schemes in public clouds are often very complex to measure. For example, Amazon EC2 does not only measure instance (virtual machine) type and running time but also network bandwidth, static IP idle time, size of persistent data stored in the cloud, etc. This is very important when running scientific computing experiments in the cloud as the size and parameters of the calculations are expected to change frequently and thus may require different amounts of computing resources every time. For this reason, it is important to estimate the computing resources needed for new experiments based on the input size and parameters used, and to estimate the cost of the deployment based on the approximated calculation time. However, the scale up parameters can only be estimated on case by case basis. We are currently in the process of developing a feasible solution using the linear programming model. Apart from this, as already mentioned in the related work section, we are also investigating how to integrate

the multi-cloud deployment of CloudMF into D2CM.

Acknowledgment

This research is supported by European Commission via the REMICS project (FP7-257793), Estonian Science Foundation grants ETF9287 and ETF8357, Estonian IT Academy, European Regional Development Fund through the EXCS and Target Funding theme SF0180008s12. The authors would also like to thank Chris Willmore and Ulrich Norbistrath who are involved with the initial development of the D2CM tool.

References

- [1] Chef. <http://www.getchef.com/chef/>, November 2015.
- [2] Libguestfs, a toolkit for modifying virtual machine disk images. <http://libguestfs.org/>, November 2015.
- [3] Libvirt virtualization toolkit. <http://libvirt.org/>, November 2015.
- [4] Press release: The 1998 nobel prize in chemistry. http://www.nobelprize.org/nobel_prizes/chemistry/laureates/1998/press.html, November 2015.
- [5] Press release: The 2013 nobel prize in chemistry. http://www.nobelprize.org/nobel_prizes/chemistry/laureates/2013/press.html, November 2015.
- [6] Amazon Inc. Amazon cloud formation. <http://aws.amazon.com/cloudformation/>, November 2015.
- [7] Amazon Inc. Amazon cloud pricing. <http://aws.amazon.com/ec2/pricing/>, November 2015.
- [8] Amazon Inc. Amazon elastic compute cloud (amazon ec2). <http://aws.amazon.com/ec2/>, November 2015.
- [9] Amazon Inc. Amazon simple storage service. <http://aws.amazon.com/s3/>, November 2015.
- [10] Amazon Inc. Amazon virtual machine import and export service. <http://aws.amazon.com/ec2/vmimport/>, November 2015.
- [11] Amazon Inc. Enabling user provided kernels in amazon ec2. <http://aws.amazon.com/articles/3967>, November 2015.
- [12] A. B. Anderson. Insights into electrocatalysis. *Phys. Chem. Chem. Phys.*, 14(4):1330–1338, 2012.
- [13] S. R. Bahn and K. W. Jacobsen. An object-oriented scripting interface to a legacy electronic structure code. *Comput. Sci. Eng.*, 4(3):56–66, 2002.

- [14] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. *SIGOPS Oper. Syst. Rev.*, 37(5):164–177, 2003.
- [15] L. S. Blackford, J. Choi, A. Cleary, E. D’Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users’ Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.
- [16] P. Blaha, K. Schwarz, G. Madsen, D. Kvasnicka, and J. Luitz. *WIEN2k, An Augmented Plane Wave + Local Orbitals Program for Calculating Crystal Properties*. Karlheinz Schwarz, Techn. Universität Wien, Austria, 2001.
- [17] C. Bunch, B. Drawert, and M. Norman. MapScale: A Cloud Environment for Scientific Computing. Technical report, University of California, Computer Science Department, 2009.
- [18] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *software Pract. Exper.*, 41(1):23–50, 2011.
- [19] F. Calle-Vallejo and M. T.M. Koper. First-principles computational electrochemistry: Achievements and challenges. *Electrochim. Acta*, 84:3–11, 2012.
- [20] Center for Atomic-scale Materials Design. GPAW. <http://wiki.fysik.dtu.dk/gpaw/>, November 2015.
- [21] CloudSwitch, Inc. Cloudswitch. <http://www.cloudswitch.com/>, November 2015.
- [22] B. E Conway. *Electrochemical supercapacitors: scientific fundamentals and technological applications*. Springer, 1999.
- [23] F. Dommert, K. Wendler, R. Berger, S. L. Delle, and C. Holm. Force fields for studying the structure and dynamics of ionic liquids: A critical review of recent developments. *ChemPhysChem*, 13(7):1625–1637, 2012.
- [24] N. Duro, R. Santos, J. Lourenço, H. Paulino, and J. Martins. Open virtualization framework for testing ground systems. In *PDATAD*, pages 67–73. ACM, 2010.
- [25] J.-P. Ebejera, S. Fulla, G. M. Morrisa, and P. W. Finn. The emerging role of cloud computing in molecular modelling. *J. Mol. Graph. Model.*, 44:177 – 187, 2013.
- [26] J. Enkovaara, C. Rostgaard, J.J. Mortensen, M. Kuisma, and et al. Electronic structure calculations with GPAW: a real-space implementation of the projector augmented-wave method. *J. Phys. Condens. Mat.*, 22(25):253202, 2010.
- [27] Eucalyptus Systems Inc. Eucalyptus. <http://www.eucalyptus.com>, November 2015.

- [28] M. V. Fedorov and A. A. Kornyshev. Ionic liquids at electrified interfaces. *Chem. Rev.*, 114(5):2978–3036, 2014.
- [29] N. Ferry, F. Chauvel, A. Rossini, B. Morin, and A. Solberg. Managing multi-cloud systems with cloudmf. In *Proceedings of the Second Nordic Symposium on Cloud Computing #38; Internet Technologies*, NordiCloud ’13, pages 38–45, New York, NY, USA, 2013. ACM.
- [30] FutureGrid. Using IaaS Clouds on FutureGrid. <https://portal.futuregrid.org/using/clouds>, November 2015.
- [31] P. Giannozzi, S. Baroni, and et al. Quantum espresso: a modular and open-source software project for quantum simulations of materials. *J. Phys. Condens. Mat.*, 21(39):395502 (19pp), 2009.
- [32] GigaSpaces Technologies. Cloudify. <http://www.gigaspaces.com/cloudify-cloud-orchestration/>, November 2015.
- [33] M. Gnahn, C. Müller, R. Répánszki, T. Pajkossy, and D. M. Kolb. The interface between Au(100) and 1-butyl-3-methyl-imidazolium-hexafluorophosphate. *Phys. Chem. Chem. Phys.*, 13:11627–11633, 2011.
- [34] M. Gnahn, T. Pajkossy, and D.M. Kolb. The interface between Au(111) and an ionic liquid. *Electrochim. Acta*, 55(21):6212–6217, 2010.
- [35] Helix Nebula. The Science Cloud. <http://www.helix-nebula.eu/>, November 2013.
- [36] High Performance Computing for the West of Scotland. The ARCHIE-WeSt service for industry users, November 2015.
- [37] V. Ivaništšev, K. Kirchner, T. Kirchner, and M. V. Fedorov. Restructuring of the electrical double layer in ionic liquids upon charging. *J. Phys. Condens. Mat.*, 27(10):102101, 2015.
- [38] V. Ivaništšev, V. Tripković, and J. Rossmeisl. Density functional theory study of a charged Au(111) EMImBF₄ (ionic liquid) interface. in prep., 2015.
- [39] K. Jorissen, W. Johnson, F.D. Vila, and J.J. Rehr. High-performance computing without commitment: SC2IT: a cloud computing interface that makes computational science available to non-specialists. In *2012 IEEE 8th International Conference on E-Science (e-Science)*, pages 1–6, 2012.
- [40] H. Kurig, M. Vestli, A. Jänes, and E. Lust. Electrical double layer capacitors based on two 1-Ethyl-3-Methylimidazolium ionic liquids with different anions. *Electrochem. Solid State Lett.*, 14(8):A120–A122, 2011.
- [41] A. Lewandowski and M. Galinski. Carbon-ionic liquid double-layer capacitors. *J. Phys. Chem. Sol.*, 65(2-3):281–286, 2004.
- [42] Q. Li and Y. Guo. Optimization of resource scheduling in cloud computing. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2010 12th Int. Symp. on*, pages 315–320, 2010.

- [43] E. Lust, G. Nurk, A. Jänes, M. Arulepp, P. Nigu, P. Möller, S. Kallip, and V. Sammelselg. Electrochemical properties of nanoporous carbon electrodes in various nonaqueous electrolytes. *J. Solid State Electrochem.*, 7(2):91–105, 2003.
- [44] M. Armbrust et al. Above the clouds: A berkeley view of cloud computing. Technical report, University of California, 2009.
- [45] M. A. L. Marques, M. J. T. Oliveira, and T. Burnus. Libxc: A library of exchange and correlation functionals for density functional theory. *Comp. Phys. Commun.*, 183(10):2272–2281, 2012.
- [46] Florian octo Forster. Collectd, a daemon for collecting system performance statistics, November 2015.
- [47] G.-B. Pan and W. Freyland. 2D phase transition of PF₆ adlayers at the electrified ionic liquid/Au(111) interface. *Chem. Phys. Lett.*, 427(1-3):96–100, 2006.
- [48] Puppet Labs. Puppet. <http://puppetlabs.com/solutions/cloud-management>, November 2015.
- [49] Racemi. Racemi cloud path. <http://www.racemi.com/cloud-path>, November 2015.
- [50] J. J. Rehr, J. J. Kas, F. D. Vila, M. P. Prange, and K. Jorissen. Parameter-free calculations of x-ray spectra with FEFF9. *Phys. Chem. Chem. Phys.*, 12(21):5503–5513, 2010.
- [51] Rightscale Inc. Rightscale servertemplate. <http://www.rightscale.com/products/configuration-framework.php>, November 2015.
- [52] J. Rossmeisl, E. Skúlason, M. E Björketun, V. Tripković, and J. K. Nørskov. Modeling the electrified solid-liquid interface. *Chem. Phys. Lett.*, 466(1-3):68–71, 2008.
- [53] L. Siinor, C. Siimenson, K. Lust, and E. Lust. Mixture of 1-ethyl-3-methylimidazolium tetrafluoroborate and 1-ethyl-3-methylimidazolium iodide: A new potential high capacitance electrolyte for EDLCs. *Electrochem. Commun.*, 35:5–7, 2013.
- [54] Skeleton Technologies OÜ. About the SkeletonC technology. <http://www.skeletontech.com/>, November 2015.
- [55] S. N. Srirama. Desktop to cloud migration. <http://mc.cs.ut.ee/mcsite/projects/desktop-to-cloud-migration>, November 2015.
- [56] S. N. Srirama, O. Batrashev, and E. Vainikko. SciCloud: Scientific Computing on the Cloud. In *The 10th IEEE/ACM Intl Symposium on Cluster, Cloud and Grid Computing(CCGrid 2010)*, page 579, 2010.

- [57] S. N. Srirama, V. V. Ivaništšev, P. Jakovits, and C. Willmore. Direct migration of scientific computing experiments to the cloud. In *High Performance Computing and Simulation (HPCS), 2013 International Conference on*, pages 27–34. IEEE, 2013.
- [58] S.N. Srirama, O. Batrashev, P. Jakovits, and E. Vainikko. Scalability of parallel scientific applications on the cloud. *Scientific Programming*, 19(2):91–105, 2011.
- [59] Y. Su, J. Yan, M. Li, M. Zhang, and B. Mao. Electric double layer of au(100)/Imidazolium-Based ionic liquids interface: Effect of cation size. *J. Phys. Chem. C*, 117(1):205–212, 2013.
- [60] A. Tõnisoo, J. Kruusma, R. Pärna, A. Kikas, M. Hirsimäki, E. Nõmmiste, and E. Lust. In situ XPS studies of electrochemically negatively polarized molybdenum carbide derived carbon double layer capacitor electrode. *J. Electrochem. Soc.*, 160(8):A1084–A1093, 2013.
- [61] V. V. Ivaništšev, S. O’Connor, and M. V. Fedorov. Poly(a)morphic portrait of the electrical double layer in ionic liquids. *Electrochem. Commun.*, 48:61–64, 2014.
- [62] C. Vecchiola, S. Pandey, and R. Buyya. High-performance cloud computing: A view of scientific applications. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 4–16, 2009.